MANUAL DE USUARIO

EASY PROPELLER

MCI-MA-0182 | REV. 1.0

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 of. 1105, Providencia, Santiago, Chile.

+56 2 23339579 | www.olimex.cl | info@olimex.cl

Ingeniería MCI Ltda.

Luis Thayer Ojeda 0115 Oficina 1105
Providencia, Santiago, Chile

www.olimex.cl
info@olimex.cl

Tel: +56 2 23339579
Fax: +56 2 23350589

® MCI Ltda. 2014

Atención: *cambios y modificaciones hechas en el dispositivo, no autorizados expresamente por MCI, anularán su garantía.*

Código Manual: MCI–MA-0182

CONTENIDO

CONTENIDO 3
INTRODUCCIÓN 4
CARACTERÍSTICAS GENERALES 4
PARTES DEL DISPOSITIVO 5
USO 6
CARACTERÍSTICAS ELÉCTRICAS 7
CARACTERÍSTICAS MECÁNICAS 7
EJEMPLOS 8
7.1 LED On/Off 8
7.2 Sensor de línea 8
7.3 Comunicación serial 9
EXPANSIÓN ANÁLOGA, DIGITAL E I2C 10
8.1 Músculo Chip Análogo digital I/O 10
8.2 Músculo Chip I2C 10
8.3 Funciones I2C 11
8.4 Ejemplo I2C 11
HISTORIA DEL DOCUMENTO 13

INTRODUCCIÓN

Esta tarjeta posee todo el poder del chip Propeller pero en un formato más versátil lo que te permite expandir funciones de manera fácil y rápida. La Easy Propeller te permite iniciar tus proyectos sin la necesidad de armar todo el hardware base, necesario para su funcionamiento.

Easy Propeller posee entradas y salidas digitales, además cuenta con un sub conjunto de pines a través de nuestro exclusivo Músculo chip, los que pueden ser indistintamente configurados como entradas análogas o I/O digitales permitiendo una amplia gama de posibilidades para los proyectos electrónicos.

Con la Easy Popeller tú puedes programar en Spin, Asembler o código C de forma rápida y fácil usando solo el cable mini USB y el programa proporcionado por Parallax.

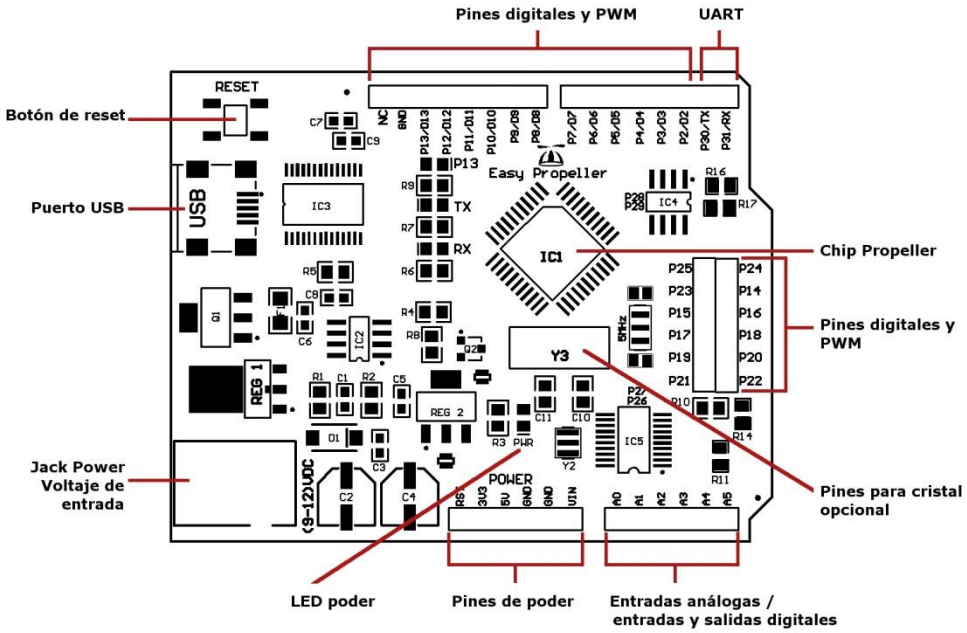
También incluye un LED conectado al pin 13 del Propeller el cual te permite tener un feedback visual o como una salida de test (digital y PWM).

CARACTERÍSTICAS GENERALES

- Alimentación y comunicación USB.
- Microcontrolador P8X32A.
- MCI Músculo chip
- Voltaje de entrada externa 7-12VDC.
- 24 pines digitales de I/O.
- 6 entradas análogas o I/O.
- 256k de memoria EEPROM.
- Reloj de 5MHz de velocidad (Se puede instalar otro cristal de mayor frecuencia).
- LED para comunicación serial (FTDI).
- LED conectado al pin 13.
- Botón de reset.
- Fusible de protección del puerto USB

PARTES DEL DISPOSITIVO

Descripción de configuraciones accesibles al usuario. Detalle modos disponibles por cada configuración.



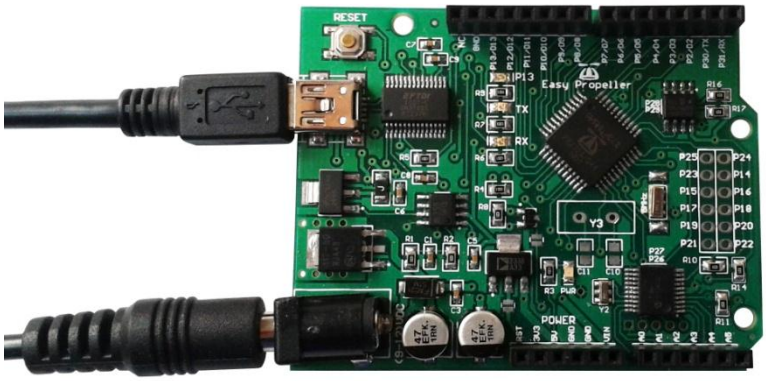
- **BOTÓN RESET:** Permite reiniciar al chip propeller.
- **PUERTO USB:** Conector mini USB el cual permite la comunicación del PC con la tarjeta.
- **JACK POWER:** Conector para alimentar la tarjeta con un voltaje externo.
- **PINES DIGITALES Y PWM:** Estos son los pines del Propeller (P2 al P13).
- **UART:** Comunicación serial del propeller.
- **CHIP PROPELLER:** Este es microcontrolador P8X32A.
- **PINES DIGITALES Y PWM EXT.:** Extensión del resto de los pines del Propeller (P14 al P25).
- **PINES PARA CRISTAL OPCIONAL:** En el caso de querer una mayor frecuencia se puede instalar un nuevo cristal (ver nota 1).
- **ENTRADAS ANÁLOGAS:** Se pueden configurar como entradas análogas como también entradas y salidas digitales.
- **PINES DE PODER:** Son los pines de los distintos voltajes junto con el reset.
- **LED POWER:** Indica si está o no alimentada la tarjeta.

USO

Para comenzar a usar esta tarjeta solo basta tener un cable mini USB y un computador con el software Propeller Tool.

Debes conectar el USB como se muestra en la siguiente figura y estás listo para comenzar a realizar tus programas.

Con esta conexión tienes alimentada la tarjeta y comunicación con el PC pero en el caso de que tus proyectos requieran mayor consumo debes conectar una fuente de 9 a12VDC con mayor amperaje (no hay problema con tener el USB y la fuente al mismo tiempo) como se muestra a continuación.



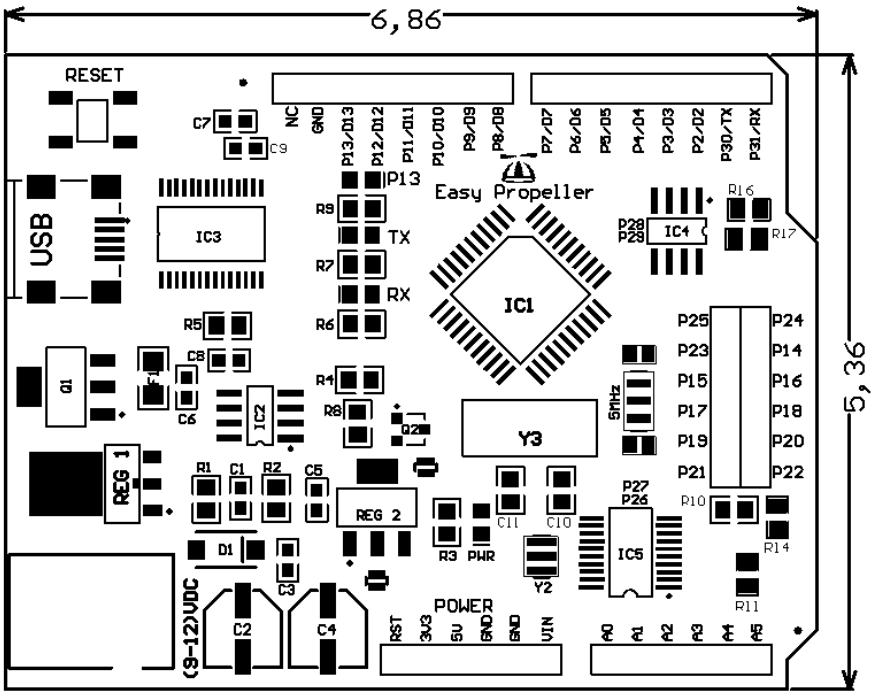
En el caso de querer usar un shield para esta tarjeta solo se debe montar siguiendo la correspondencia de pines.

CARACTERÍSTICAS ELÉCTRICAS

- Voltaje de entrada: 7VDC a 12VDC
- Regulador de voltaje de 5V
- Regulador de voltaje de 3,3V
- Consumo promedio: 50mA

CARACTERÍSTICAS MECÁNICAS

Dimensiones: 6,86cm x 5,36cm



EJEMPLOS

Usaremos el programa Propeller Tool el cual nos permite trabajar con el lenguaje SPIN. Este software se puede descargar desde nuestra página web en la descripción del producto.

Una vez instalado el software procedemos a escribir nuestros ejemplos (también se encuentran disponibles para descargar en la descripción del producto).

7.1 LED ON/OFF

En el siguiente ejemplo se prenderá y apagará el LED que viene en la placa con un retardo entre cada periodo.

```
CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  Pin = 7          '' assign constant

PUB Led_On_Off

  dira[Pin] := 1    '' configure pin with output

    repeat '' cycle repeat
      '' Configure pin with HIGH
      outa[Pin] := 1
      '' delay
      waitcnt(clkfreq/4 + cnt)
      '' Configure pin with LOW
      outa[Pin] := 0
      '' delay
      waitcnt(clkfreq/4 + cnt)
```

7.2 SENSOR DE LÍNEA

Ahora usaremos el sensor digital de línea el cual nos permitirá detectar una línea negra y apagar el LED que trae la tarjeta.

```
CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  LED = 13        '' assign constant
  SENSOR = 7      '' assign constant

PUB Main

  dira[LED]  := 1         '' configure pin with output
  dira[SENSOR] := 0       '' configure pin with input

  repeat

    outa[LED] := ina[SENSOR] '' read the sensor and pass the value to LED
```

7.3 COMUNICACIÓN SERIAL

Para este ejemplo leeremos el sensor de línea y enviaremos el estado al PC y para ello usaremos la librería "FullDuplexSerial" de la empresa Parallax Semiconductor.

```
CON
  _clkmode = xtal1 + pll16x
  _xinfreq = 5_000_000

  SENSOR = 7                        '' assign constant

VAR

  LONG valor                        '' assign variable

OBJ
  Serial  : "FullDuplexSerial"      '' library of Parallax Semiconductor

PUB Main

  dira[SENSOR] := 0                '' configure pin with input
  Serial.start(31, 30, 0, 9600)    ''start(rxPin, txPin, mode, baudrate)

  repeat
    valor := ina[SENSOR]           '' read the sensor and pass the value to valor
    Serial.str(string("El valor del sensor es: "))  '' write in serial port
    Serial.Dec(valor)                               '' write valor in serial port
    Serial.tx(13)
    waitcnt(clkfreq/4 + cnt)                        ''delay
```

Para mayor información sobre el microcontrolador visita el sitio web de Propeller www.parallax.com/propeller/ en donde podrás descargar toda la información junto con el software para programar.

También puedes ver ejemplos de programación en lenguaje C y C++ en https://sites.google.com/site/propellergcc/examples donde nos muestran cómo usar el puerto serial.

EXPANSIÓN ANÁLOGA, DIGITAL E I2C

Esta tarjeta posee un chip pre-programado llamado Músculo que está conectado al microcontrolador Propeller y su función es expandir potencialidades de conversión análogo digital y expansión I/O digital sobre los pines A0, A1, A2, A3, A4 y A5 de la tarjeta Easy Propeller, esto potencia al Propeller entregandole un complemento óptimo para él.

Para hacer uso de esta expansión se entrega una librería llamada ADC_IO_MOD.spin donde contiene todas las instrucciones de configuración y uso.

8.1 MÚSCULO CHIP ANÁLOGO DIGITAL I/O

En la siguiente tabla se muestra los comandos de la librería que se tiene que usar con el Propeller para configurar el Músculo chip y también para el uso de los pines de expansión.

Comando	Descripción
start	Inicializa el módulo
SetAnalog(Pin)	Set un pin (A0..A5) como entrada análoga
SetDigitalOutput(Pin)	Set un pin (A0..A5) como salida digital
ReadInputPin(Pin)	Set un pin (A0..A5) como entrada digital
ReadAnalog(Pin)	Lee el valor análogo
SetPin_LOW(Pin)	Salida del pin en estado bajo
SetPin_HIGH(Pin)	Salida del pin en estado en alto
ReadInputPin(Pin)	Lee el valor digital

8.2 MÚSCULO CHIP I2C

Musculo chip posee dentro de sus prestaciones la implementación del bus de comunicación I2C para dispositivos externos ej: eeprom,rtc,sensores etc . Está opción asigna los pines **A4** como **SDA** y **A5** como **SCL** y le ofrece al programador tres sencillas funciones que lo liberan de la necesidad de conocer el protocolo I2C.

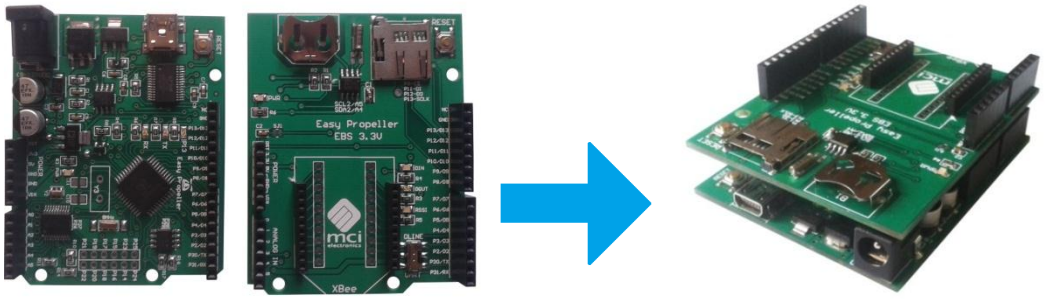
8.3 FUNCIONES I2C

Comando	Descripción
I2C_Begins	Implementa la comunicación I2C sobre los pines A4 y A5
I2C_Writes(address I2C device, register, value)	Escribe un valor en el dispositivo
I2C_Reader(address I2C device, register)	Lee el valor del dispositivo

8.4 EJEMPLO I2C

Ahora daremos un ejemplo donde se conecta la Easy Propeller con su shield la Easy Propeller EBS ya que tiene el RTC DS1307 que se comunica por I2C a través del Músculo chip , donde programaremos una hora y luego en un loop leer la hora del RTC y mostrarla por el puerto serial.

Para el montado se debe de realizar con precaución coincidiendo los pines correctamente como se muestra en la siguiente imagen a continuación.



En el caso de cortar la energía el reloj de tiempo real sigue funcionando siempre y cuando esté instalada la pila de 12mm (MCI-PRT-00677).

```
CON
        _clkmode = xtal1 + pll16x
        _xinfreq = 5_000_000

        RTC_ADDRESS = $D0   'I2c address RTC DS1307
        'RTC Registers
        RTC_SECONDS = $00
        RTC_MINUTES = $01
        RTC_HOURS  = $02
        RTC_DAY = $03
        RTC_DATE = $04
        RTC_MOUNT = $05
        RTC_YEAR = $06

VAR
  byte minute
  byte hour
  byte seconds

OBJ
 extMod : "ADC_IO_MOD.spin"
 Serial: "FullDuplexSerial"
PUB Main

  Serial.start(31, 30, 0, 9600)
  extMod.start
  extMod.I2C_Begins
  extMod.I2C_Writes(RTC_ADDRESS,RTC_SECONDS,$00)
  extMod.I2C_Writes(RTC_ADDRESS,RTC_MINUTES,$00)
  extMod.I2C_Writes(RTC_ADDRESS,RTC_HOURS,$12)
  repeat
        Serial.str(STRING("TIME = "))
        hour:= extMod.I2C_Reader(RTC_ADDRESS,RTC_HOURS)
        minute:= extMod.I2C_Reader(RTC_ADDRESS,RTC_MINUTES)
        seconds:= extMod.I2C_Reader(RTC_ADDRESS,RTC_SECONDS)
        Serial.hex(hour,2)
        Serial.str(STRING(":"))
        Serial.hex(minute,2)
        Serial.str(STRING(":"))
        Serial.hex(seconds,2)
        Serial.tx(13)
        Serial.tx(10)
```

Recuerda que esta tarjeta acepta solo niveles de 3,3V por tanto debes instalar shield que trabajen con ese voltaje.

Cada vez que se conecta el puerto USB el Propeller se reinicia, incluso cuando esta se encuentra alimentada por el jack power.

HISTORIA DEL DOCUMENTO

Revisión	Fecha	Editado por	Descripción/Cambios
1.0	20 de febrero de 2013	Diego Muñoz O.	Versión inicial del documento